DATALOGIC™

DLReader

User's Manual

Datalogic Automation S.r.l.
Via Lavino, 265
40050 - Monte S. Pietro
Bologna - Italy

DLReader User's Manual

Ed.: 09/2011

© 2011 Datalogic Automation S.r.l. □ ALL RIGHTS RESERVED. □ Protected to the fullest extent under U.S. and international laws. Copying, or altering of this document is prohibited without express written consent from Datalogic Automation S.r.l.

Datalogic and the Datalogic logo are registered trademarks of Datalogic S.p.A. in many countries, including the U.S.A. and the E.U.

Matrix 210, Matrix 410, ID-NET, VisiSet and X-PRESS are trademarks of Datalogic Automation S.r.l. All other brand and product names mentioned herein are for identification purposes only and may be trademarks or registered trademarks of their respective owners.

Datalogic shall not be liable for technical or editorial errors or omissions contained herein, nor for incidental or consequential damages resulting from the use of this material.

13/09/11

CONTENTS

1 BASIC USE OF DLREADER 5
1.1 Loading an Image 5
1.2 Creating a ROI 5
1.3 OCR Configuration Parameters Setup 7
1.3.1 Standard Configuration 7
1.3.2 Automatic Configuration 7
1.3.3 Advanced Configuration 8
1.4 Font Training 9
1.5 Unrecognized Character (Sub Char) 11
1.6 Font Training to Speed Up OCR 11
1.7 Train Patterns 12
1.8 Copy Trained Font 13
1.8.1 Sample of Support Font Files in the File System 14
1.9 Match String 15
1.10 Acceptance, Fail and Warn (Quality Level) 16
1.11 Constraints on the OCR Configuration 17

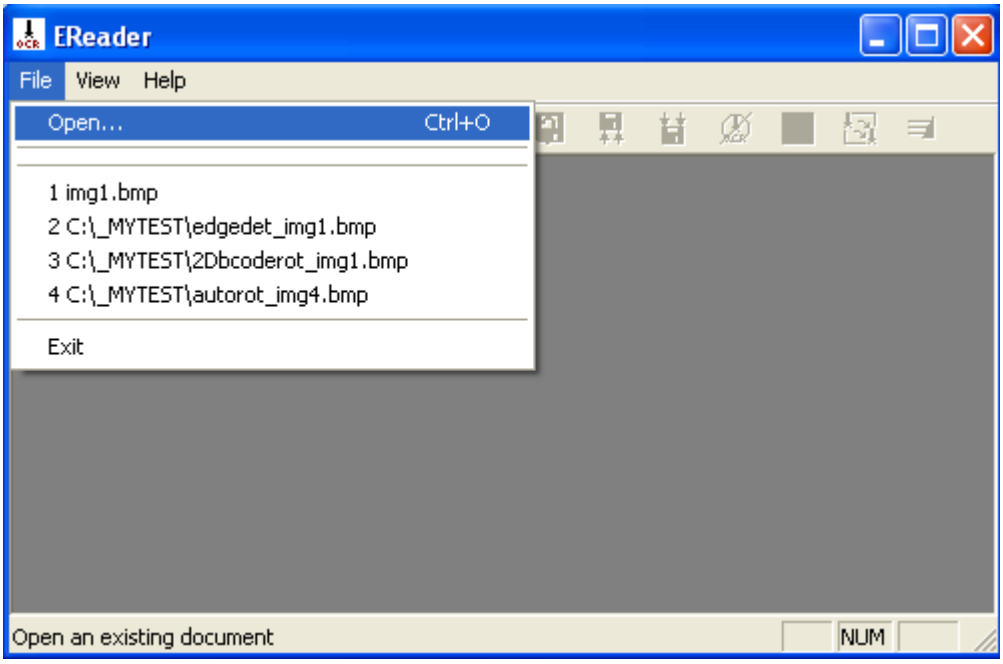
2 DLREADER PARAMETERS 18
2.1 Reader and Character Constraints 18
2.1.1 Aspect Ratio Independence 21
2.1.2 Character Size and Character Spacing 21
2.1.3 Character Scoring 21
2.2 Threshold 22
2.3 OCV Option —NOT USED 23
2.4 Fielding —NOT USED 24
2.5 Training 25
2.6 Font File and Font List Editor 25
2.7 Image Option 28
2.8 Auto Configuration 29
2.9 Saving and Restoring Reader Settings 30
2.10 Changing Images 30

3 EDGE FINDERS 32
3.1 CREATING an EDGE ROI 32
3.2 Finding the Edge 33
3.3 Output Information (shown only graphically) 33

1 BASIC USE OF DLREADER

1.1 LOADING AN IMAGE

The command “File”-> “Open” loads the bitmap.



1.2 CREATING A ROI

With the command “OCR”-> “New Reader” a generic ROI is created.

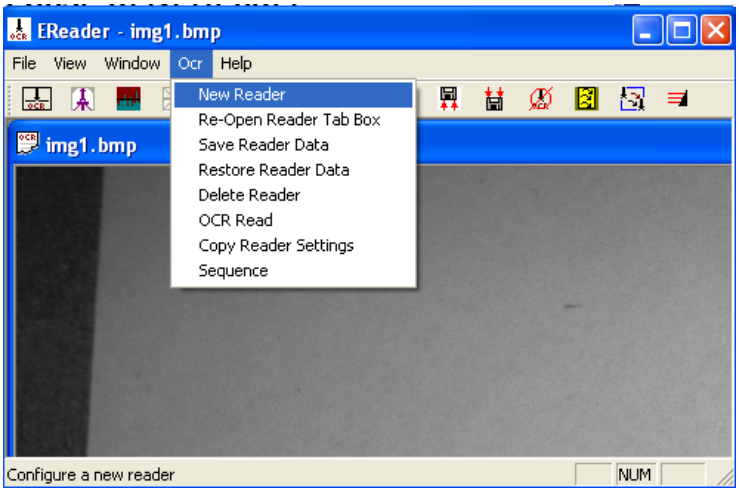
- ⇒ The user is prompted to choose a name (unique) to give to the ROI.
- ⇒ The user’s needs to correctly position it on the string to be recognized.

NOTE:
The ROI cannot contain strings placed on multiple lines; if more lines should be read, a new ROI should be created for each line.

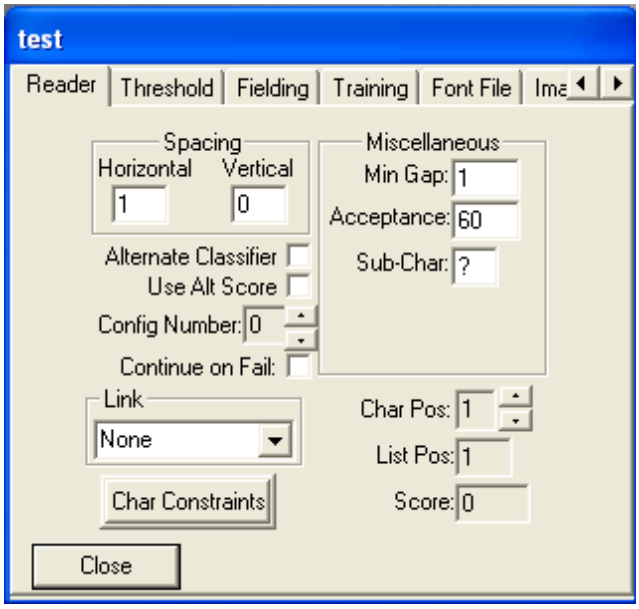
The ROI is better if it contains only the string to be read (to avoid parts of other strings or parts of designs).

If the ROI contains strings on the same line separated by spaces, the spaces will be ignored and the strings will be concatenated into a single output. This should create a number of ROI equal to the number of strings to be read even if they are on the same line.

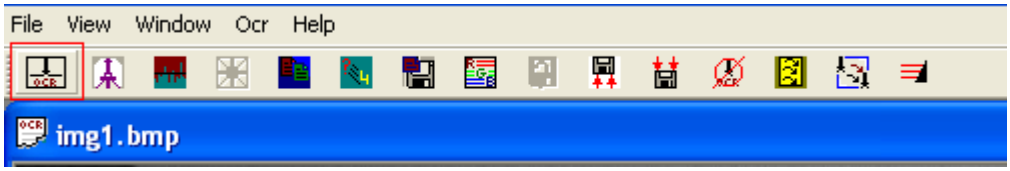
The only supported ROIs are rectangular and can be configured to find only horizontal/ vertical strings.



In addition to the rectangle to be positioned, it also opens a configuration window.



1.3 OCR CONFIGURATION PARAMETERS SETUP



There are three configuration levels of the character segmentation:

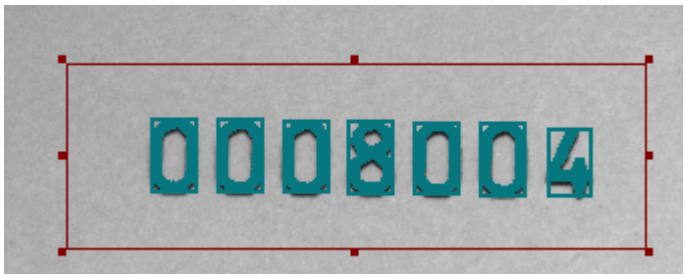
- ⇒ **Standard Configuration** – this is the first to be tried. It works well if the characters are sufficiently separated and the contrast is good.
- ⇒ **Automatic Configuration** – this is the second to be tried and solves many types of difficult applications.

Advanced Configuration – this requires a thorough knowledge of the DLReader tool. Please refer to the documentation included with the tool.

Each time you press the button , the segmentation of the characters on the configuration chosen is highlighted in the selected ROI.

1.3.1 Standard Configuration

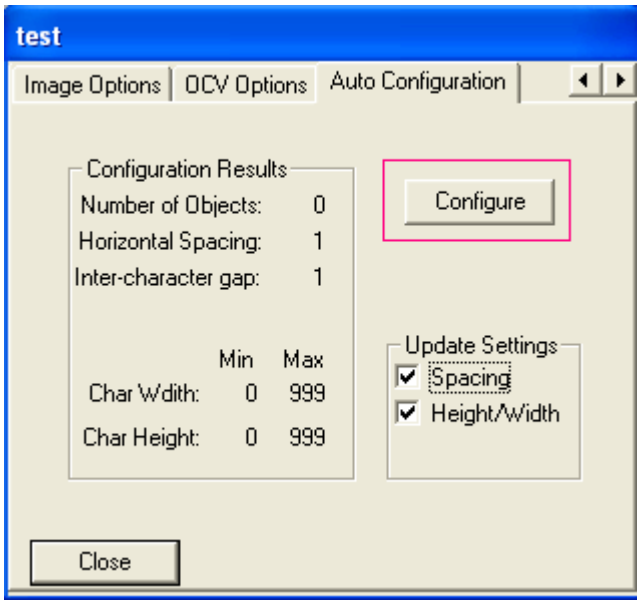
For a ROI like the one in the figure, the standard configuration is usually fully satisfactory. Each character has in fact its own rectangle distinct from others.
You can proceed with the Font training.



1.3.2 Automatic Configuration

For a ROI like that in the figure, the standard configuration is not enough: the characters ‘t’ and ‘z’ have been boxed together.

You can try the automatic configuration by pressing the “Configure” button in the Auto Configuration tab of the configuration window.



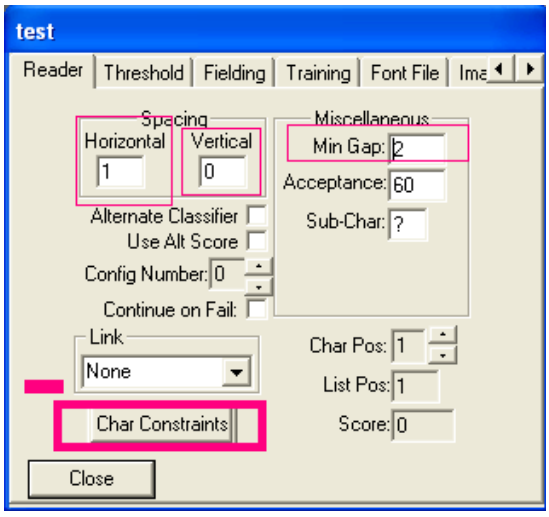
At this point you must try the automatic configuration of characters, still using the button .

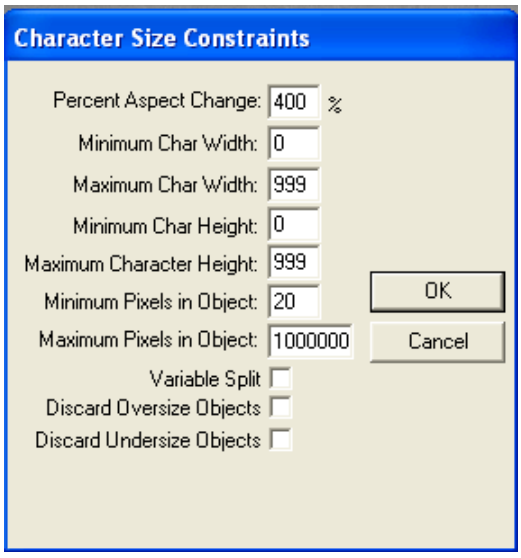
If the result is good, you can proceed with the Font training, otherwise go to the Advanced configuration.

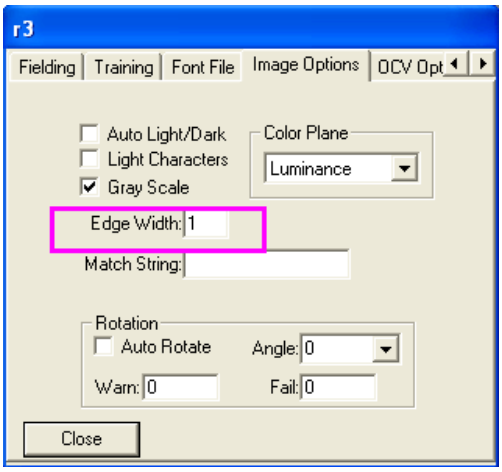
1.3.3 Advanced Configuration

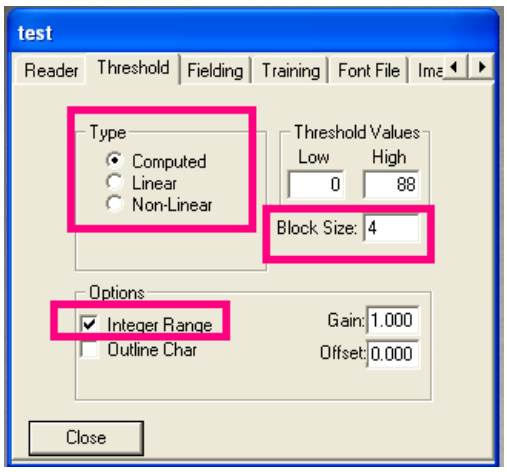
The parameters that are most often changed in the advanced configuration are:

- Horizontal/Vertical Spacing
- Min Gap
- Char Constraints (see window II alongside)
- Edge Width is very useful to make the segmentation more robust.
- If necessary, also the type of segmentation used.



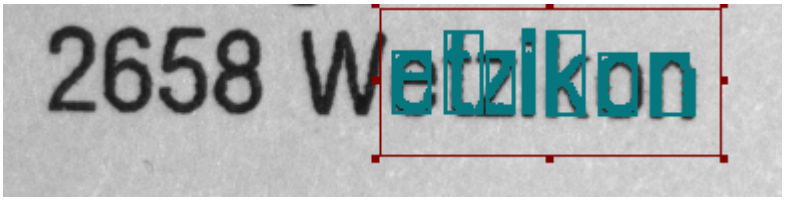






Using these parameters has a direct effect (NEGATIVE) on the computing time and the scaling capacity of the OCR.

One possible segmentation of the ‘t’ and ‘z’ is shown in the picture but it is not very satisfactory.

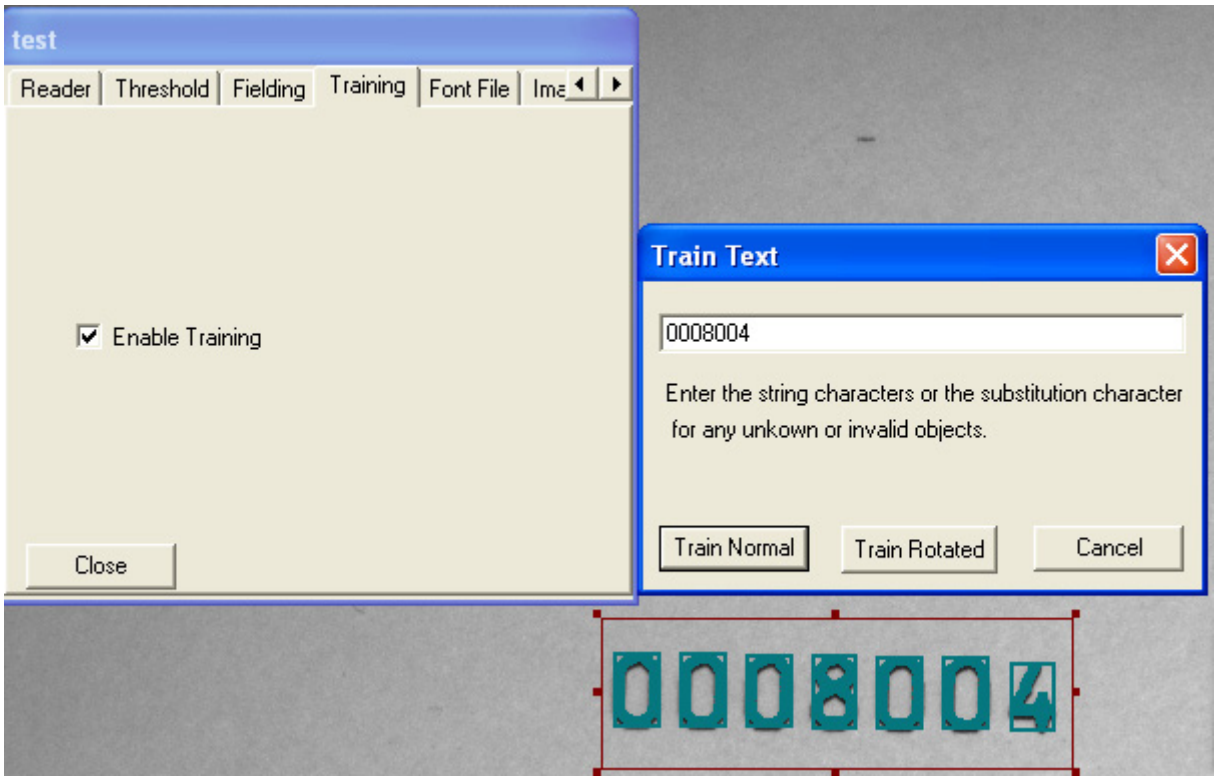


1.4 FONT TRAINING

Once a correct segmentation of the characters has been made, you have to run font train.
You must put a checkmark in the Enable Training box on the Training tab in the configuration window.

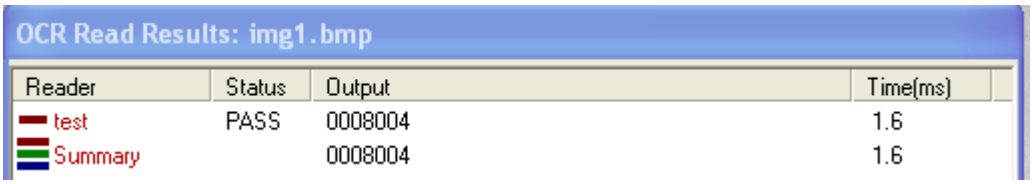
Press the OCR button of the DLReader.
This opens a pop up that shows the characters making up the string.

To run the training, press “Train Normal.”



Now you must test the reading by pressing the button .

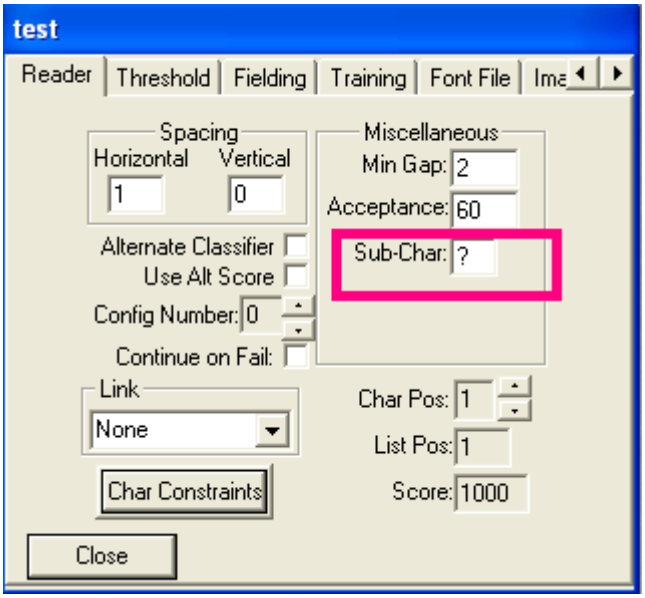
The result is shown in the window OCR Read Results.



Note that the “space” is not a character, it cannot be segmented and it shouldn’t be present in the trained string.

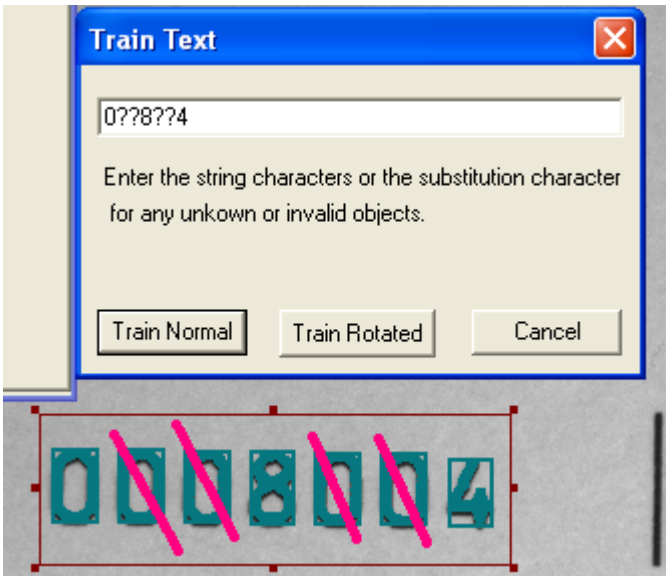
1.5 UNRECOGNIZED CHARACTER (SUB CHAR)

The character that should be displayed when a character string is NOT recognized is configurable in the configuration window. The default is the character '?'.
It must be a character that does NOT belong to the alphabet of the Font.



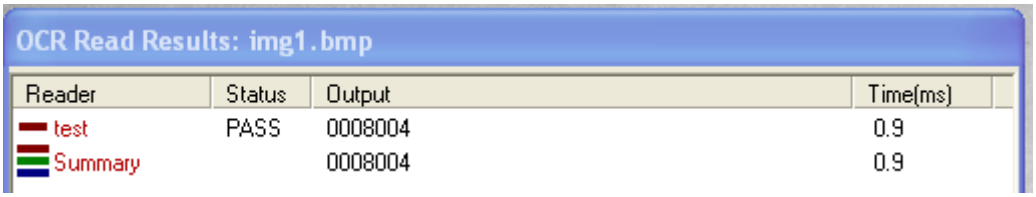
1.6 FONT TRAINING TO SPEED UP OCR

Not all the characters have to be entered. Actually, for faster recognition, it's better to start by entering only a part of each type. When you do NOT want to run the training for a character in the training string, replace it with the character of sub-char (by the default '?').



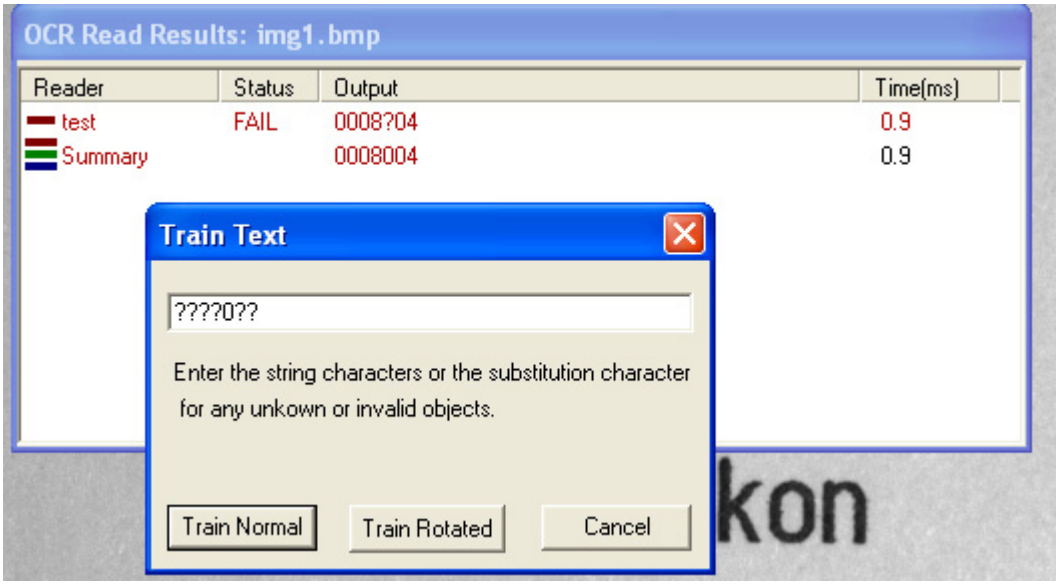
That way the Font is composed of one element of type 0, an element of type 8, and an element of type 4 (instead of 5 elements of type 0, a type 8, and a type 4).

At this point the reading must be tested by pressing the button .
The result is shown in the window OCR Read Results.



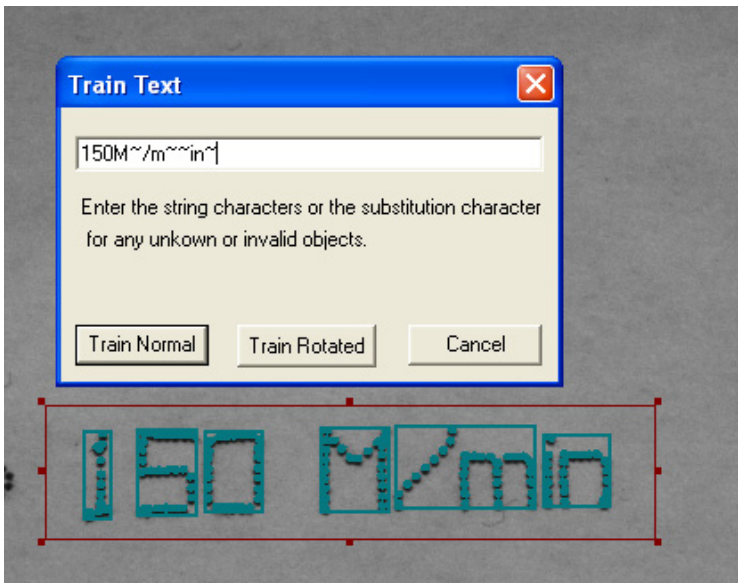
In this case, all the characters were recognized correctly.
If this had not been so, it would have been necessary to proceed by running the train for unrecognized characters.

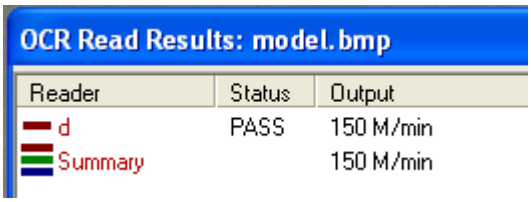
For example if the character 0 in fifth position was not recognized, the new string in the train would be “????0??”.



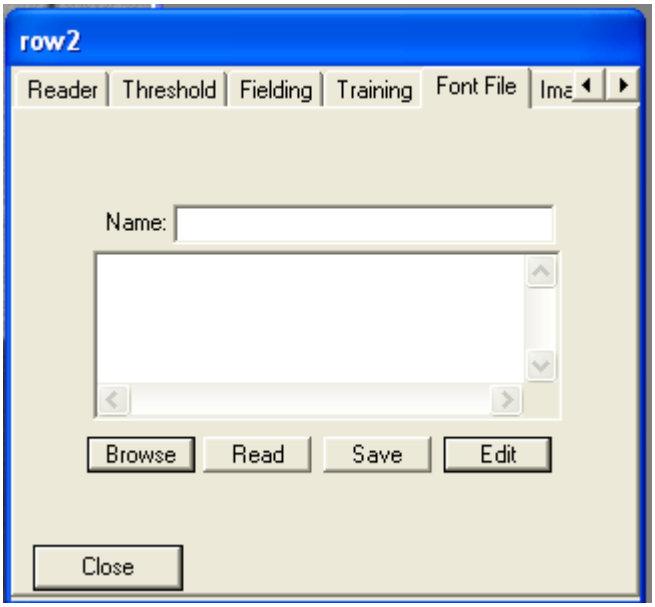
1.7 TRAIN PATTERNS

In case of group of chars that is hard to split, it is possible to train the entire pattern using the special char “~” before and after.

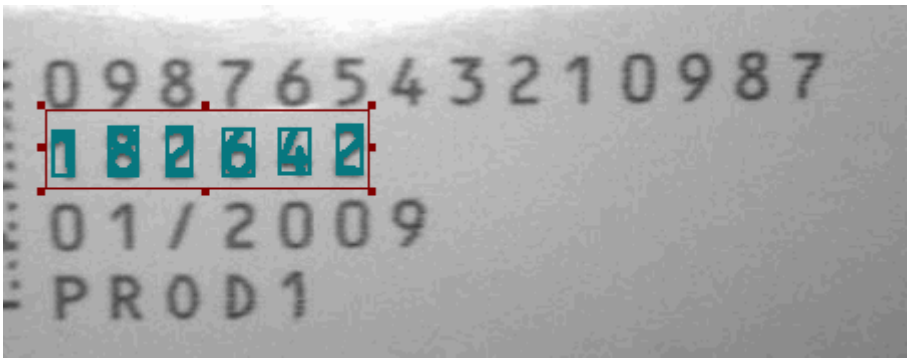




1.8 COPY TRAINED FONT

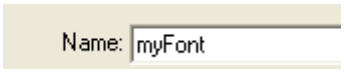


There is a fast way to use the same font on multiple rows without having to train it again. In general this operation is not recommended because it is better to train the font for each line but is still useful to know how it works.



Suppose you have trained a complete font for line 2 (reader row2). Now you may use the same font for line 3.
The steps are as follows:

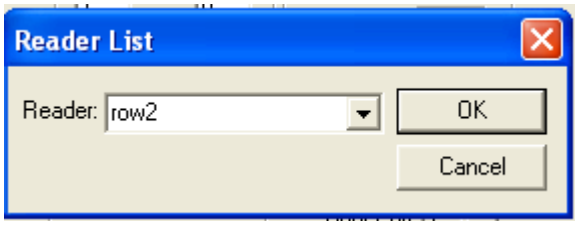
In the Font Files tab of Row2 :



write a name on the Name line (eg., “myFont)
then SAVE. When prompted, “Delete font file in memory?”press Yes
then run a READ. When asked, “Delete font file in memory?”press Yes

Create a new reader Row3.

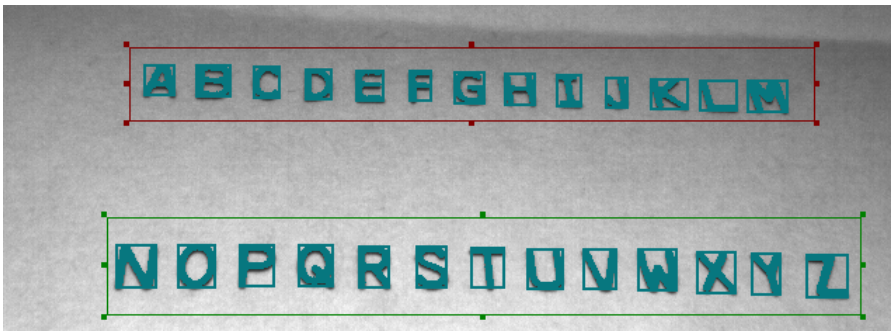
Selecting the new reader Row3, copy the parameters of reader Row2 by pressing the appropriate key (Copy reader setting). Choose row2 as the reader to be copied.



In the Font Files tab of Row3 make sure that there is the same name used in row2 Name row (eg., “myFont).
Before saving the DLR file, you can delete the names of the various Font File tabs (so the font is copied from RAM).

Note: This process can be very useful when creating the Font if you want to use multiple readers to create a single Font. The file Name is then used as partial support of the Font.

1.8.1 Sample of Support Font Files in the File System



To make a complete train of the alphabet from A to Z AND use it the ROI in both, there are two methods:

METHOD 1:

- Create a Reader AM
- make the Font train ABCDEFGHIJKLM,
- move the ROI AM down
- make the Font train NOPQRSTUVWXYZ

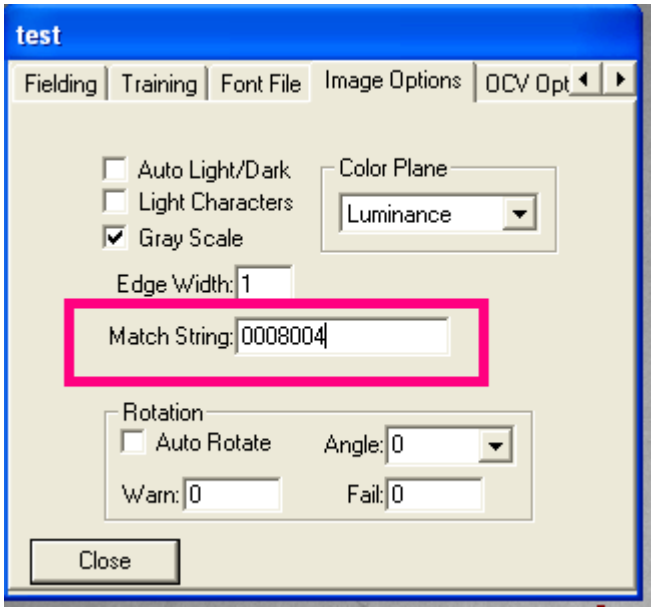
- then follow the method above to create the NZ reader and associate it with the same AM font:
    - In the AM Font Files tab: Write a name in the Name line (eg., “myFont”) SAVE and run a READ.
    - Create a new NZ reader. Copy the parameters of the AM reader in NZ.

METHOD 2 :

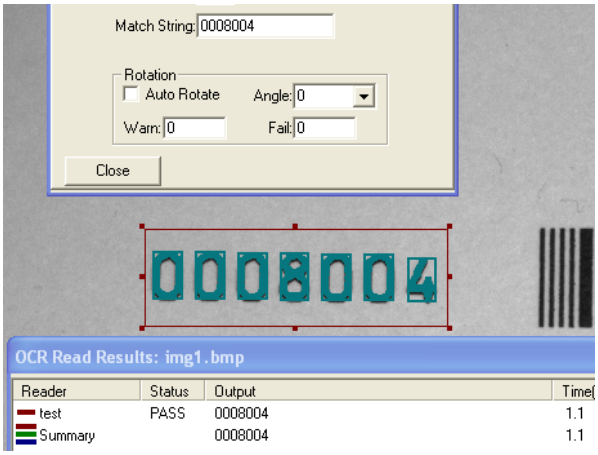
- Create the AM Reader; make the Font train ABCDEFGHIJKLM.
- In the AM Font Files tab: Write a name in the Name line (eg., “myFont”) SAVE. and run a READ.
- Create an NZ reader; Copy the parameters of the AM reader in NZ. Make the Font train NOPQRSTUVWXYZ. SAVE and run a READ.
- Reopen the AM and run a READ (to update your Font with the new characters)

1.9 MATCH STRING

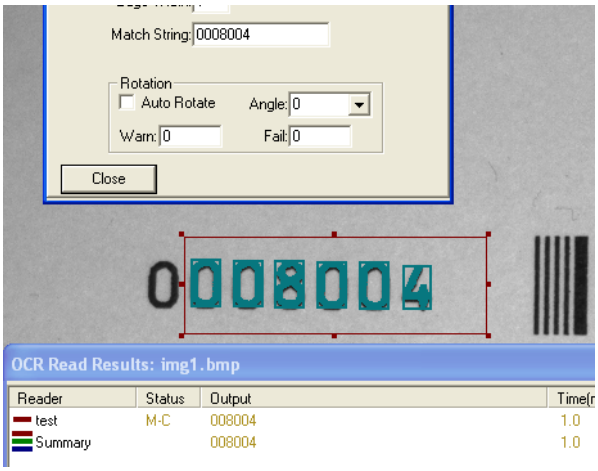
An expected string can be configured if it is known.



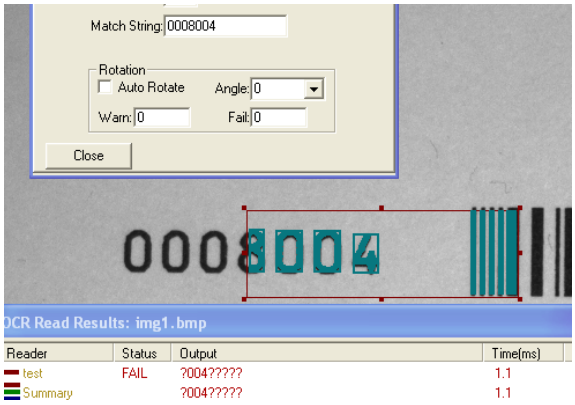
The status of the OCR Read result window will show **PASS** in the case of a match.



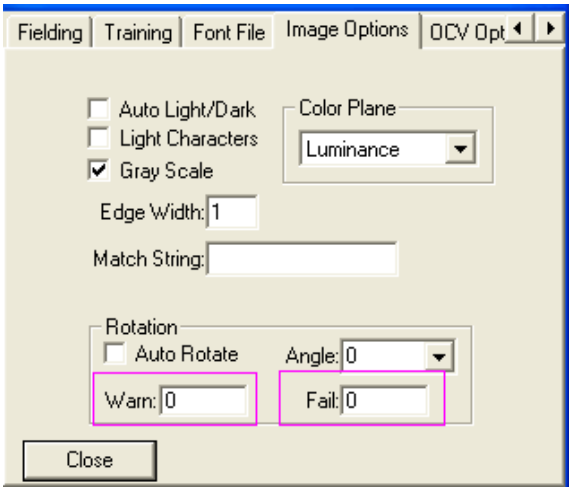
The status of the OCR Read result window will show **M-C error** if NOT a match (in this case the ROI was moved).
All the characters were recognized as belonging to the trained Font but the sequence is NOT the same as indicated.

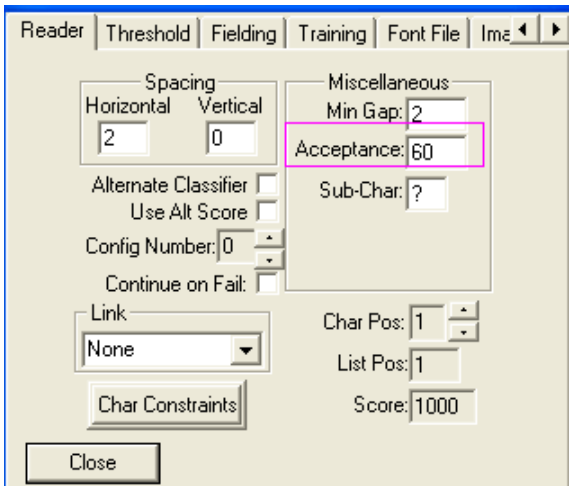


The status of the OCR Read result window will show **FAIL error** when NOT a match and also for characters NOT recognized (in this case the ROI was moved).



1.10 ACCEPTANCE, FAIL AND WARN (QUALITY LEVEL)





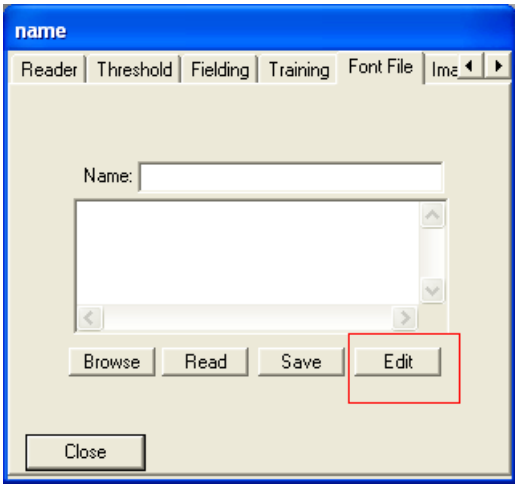
Acceptance
Typical values range from 60-75.
The minimum value is 1. The maximum value is 100.
The value 0 implies a special use of ROI as a locator.
If the Fail parameter is not set (Fail default is 0), it is the only cause that make a char not recognized.

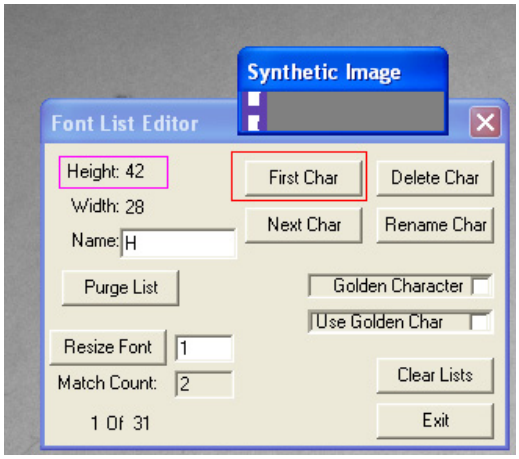
Fail
Fail is the parameter defining the required quality level (acceptable) in character recognition.
Typical values range from 700-1000.
The minimum value is 1. The maximum value is 1000.
When the Font train runs, it is good to set it to 900.
If it's value is not 0, the char's score is compared with this value and an error char ('?') is returned instead of the recognized char if the score is lower.

Warning
Warn parameter defines a second level of quality level in character recognition.
Typical values range from 700-1000.
The minimum value is 1. The maximum value is 1000.
It is usually greater than the Fail value.
If it's value is not 0, the char's score is compared with this value and a warning error is returned by the library to the application SW if the score is lower.

1.11 CONSTRAINTS ON THE OCR CONFIGURATION

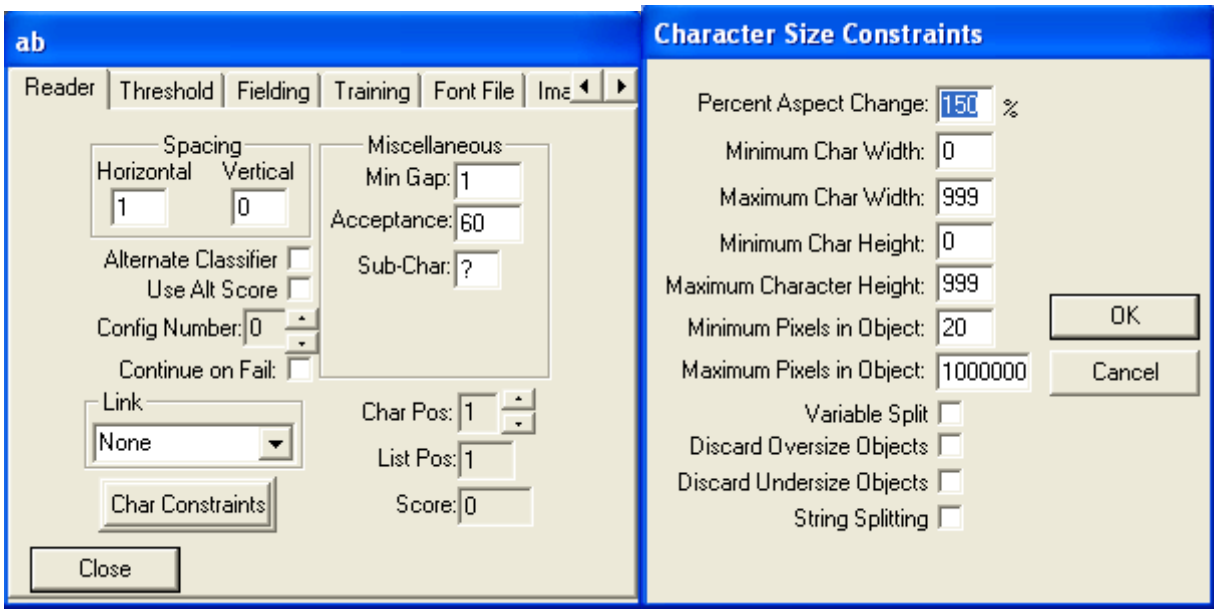
The characters of the string to be recognized must have an average height in the range [28.45] of pixels.
Characters are allowed above or below average height, but the reliability and recognition times are not optimal.
After running the Font train on DLReader, the font can be controlled by pressing the Edit button in the Font File tab.
The Font List Editor opens. Pressing First Char (and then Next Char until the end of the Fonts) you can see the image of the character training (Synthetic Image), its name (Name) and its dimensions of height and width in pixels (Height, Width) .
In this way you can verify if the height of the characters is in the required range





2 DLREADER PARAMETERS

2.1 READER AND CHARACTER CONSTRAINTS



By default, horizontal spacing has a value of one (1) and vertical spacing has a value of zero (0). When working with any character or object that consists of more than one piece, it may be necessary to adjust one or both of these values.

Spacing
Horizontal Spacing defines how far apart dots or pieces of a character can be located. This value is in pixels and is in the horizontal direction only. If the value is too small, not all parts of a character or object will be included. If the value is too large, adjacent characters or objects may be included. The spacing value must not be greater than the smallest distance between characters. (This restriction may be ignored when using the Max Width parameter.)

Vertical Spacing is the same as horizontal spacing except in the vertical direction. Unlike horizontal spacing, the default and minimum value is zero (0). Rather than mean no adjacent objects in the vertical direction, zero (0) means **ALL** objects in the vertical direction. With proper location of the ROI and selection of thresholding type, zero (0) will usually provide the best results. Non-zero values must be used with caution with dot matrix characters. The underlying algorithm uses this parameter to discard extraneous data above and below characters, which in turn, may cause the DLReader to ignore valid dot matrix objects. It is generally recommended to use Max Character Height (see **Char Constraints** tab) rather than using a vertical spacing value other than zero (0).

Min Gap
This defines the minimum horizontal spacing between adjacent characters. When set, the DLReader “skips” over this number of pixels before starting its search for the next character. The default value is one (1). Setting the minimum gap can improve performance times especially when the spacing between objects is very large.

Acceptance
When comparing an object against objects in the font list, the acceptance value is used as a low-end “cut off” parameter by the internal algorithms. The range for the majority of applications is between 60 and 75 when using the relaxed mode. The values may be a little lower when not using the relaxed mode. For applications concerned with both reading and quality of marking, the acceptance should be set between 75 and 85. In addition, only high quality objects should be trained since the acceptance is against trained characters. High resolution made may require necessarily lower acceptance values. Empirical testing will yield the optimal settings, with 60 being the default.

Sub-Char
When the DLReader cannot match an object with a trained object, the DLReader returns the character defined in the *Substitute Character* (Sub-Char) edit box found in the **Reader** tab page. By default it is a question mark, however, it can be any character that is not normally read.
The character in this edit box is also used during training to skip over objects that are not to be trained. Since training is by object position, some kind of placeholder must be used when an object is to be left as is. The substitute character is used for this purpose.

Alternate Classifier
Different OCR algorithm.

Use Alt Score
More accurate scoring used always for the Alternate classifier and enabling for the original OCR algorithm.

Continue on Fail
If unchecked, the reader returns anytime a failed read occurs. This can be either too few characters, as defined in the **Min Char** box, or a misread.
If checked, then the reader continues reading until all regions have been read.

Link
Readers can be linked using the Link tool on the Reader tab page. Linking readers is an act that only affects the ROI of each linked reader. It does not affect the independence of the readers. It also does not affect any readers that are not linked to that reader. Changes to the size, shape and position of a reader’s ROI will be reflected in every ROI that is linked to it. This is advantageous when the settings are used on a new image where the position of the character string is different and the ROI’s must be relocated. The linking tool is useful with the searching feature to be discussed later.

Character Constraint Parameters

Percent Aspect Change
Internally, the reader is totally size and aspect ratio independent. This means, that changes in height and/or width, regardless of amount, have little or no impact on the ability to classify characters. As a practical matter, however, it is desirable and sometimes necessary to limit the amount of variation allowed. By default, this is set to 400%. This means a newly found objects must have a height or a width within a range of one-quarter (1/4) to four (4) times the size of the stored trained character it is being compared against.

If the newly found object is outside of these limits, no comparison to that trained character is done.

Two primary benefits occur when setting this value. First, reader accuracy is improved because only objects of similar height and width are used for classification. Secondly, speed is improved because potentially fewer trained characters are used for classification.

Minimum Char Width
Setting the minimum character width enables the reader to discard any found object which has a width less than this value. This is true regardless of any other settings.

Maximum Char Width
Maximum character width defines the maximum bounding box width. If the **Variable Split** checkbox is unchecked, this is a “hard” number. That is, all objects that potentially could be wider are set to this width. If the **Variable Split** checkbox is checked, then the maximum width can vary slightly when the reader selects the “optimal” split column.
If the **Discard Oversize Objects** checkbox is checked, any object wider than this value is discarded and not used for classification.

Minimum Char Height
As it implies, this sets the minimum height that any character can be. Unlike **Minimum Character Width**, character heights are adjusted upward if the newly found object is too short. Adjustments are automatic.

Maximum Char Height
Objects that are too tall can be automatically resized by setting this value.
If the **Discard Oversize Objects** checkbox is checked, any object taller than this value is discarded and not used for classification.

Minimum Pixel in Object
This defines the minimum number of pixels within the threshold range that an object must contain. If the object has fewer pixels than this value, it is discarded and not used for classification

Maximum Pixel in Object
This defines the maximum number of pixels within the threshold range that an object must contain. If the object has a greater number of pixels than this value, it is discarded and not used for classification.

Variable Split
When checked, this enables an “intelligent” algorithm to look back a few pixels to identify the optimal split point between characters.
Note: This is not recommended for widely spaced, cleanly printed dot matrix characters.

Discard Oversize Objects
When this flag is set, objects that exceed the **maximum pixel** settings for either width or height are not considered for a match.

Discard Undersize Objects
When this flag is set, objects that are less than the **minimal pixel** settings for either width or height are not considered for a match.

String Splitting
This option is very useful for attached chars.

2.1.1 Aspect Ratio Independence

Aspect ratio independence means that objects can be taught at one size and height/width ratio and be read at a different size and/or height/width ratio. Internally, the software limits the amount of change to ±400%. As a practical matter, it should be kept to about ±50%. Not only does this provide greater flexibility when training, but also in many applications the size of the text or objects can vary due to a variety of process causes.

While this feature is beneficial in many applications, there are fonts or characters that can pose special problems when viewed in the light of aspect ratio independence. For example, what is the difference between a square period and a straight line 'l'? What is the difference between the letter 'O' and number zero '0' in an aspect ratio independent manner? The answer to both questions is that there is no difference.

2.1.2 Character Size and Character Spacing

The ideal range in height is **from 25 pixels to 40 pixels**. Smaller characters require more training due to the limited amount of data. This causes small object changes to have a large impact on the stored data, thus making them harder to interpret. Characters taller than 40 pixels do not add any additional data that would be beneficial in matching objects. Larger objects also have the side effect of being slower to read due only to their size.
As long as characters are at least one (1) pixel apart in the horizontal direction, spacing is generally not a factor. Only if the spacing is quite large does it start to impact the performance. Remember, the DLReader must check every pixel after the last object in order to find the next object. The larger the spacing and/or the taller the ROI, the greater the number of pixels to be checked. The greater the number of pixels to be checked, the more time that is required to process.

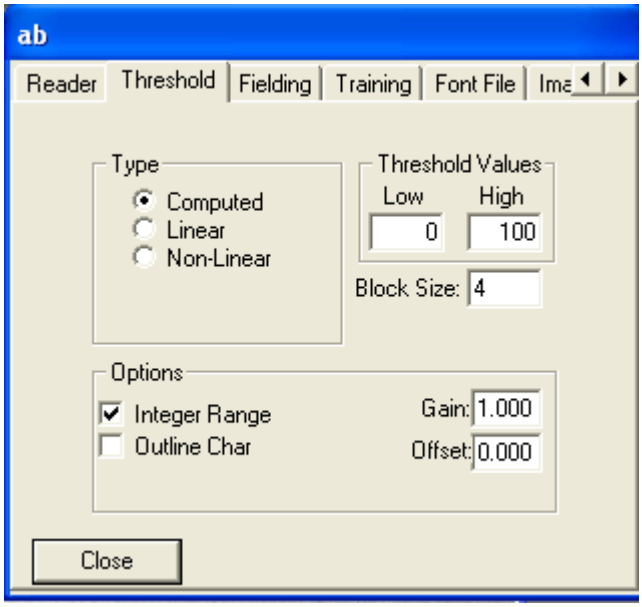
Character size, font list length, character spacing, and number of objects all contribute to the total processing time. Proper training is a key component in ensuring the minimum number of trained characters in the list.

2.1.3 Character Scoring

When any object is read, a "score" is calculated based upon how well that object conforms to the object it was matched against. A perfect score, and therefore a perfect match, is 1000. The minimum score can be less than zero in some circumstances. An object's score is listed on the Reader tab page.

Typical values range between 500 and 1000. For OCR purposes, the score is not generally used.
For setup purposes, the score is used to give the user an indication of how the DLReader is comparing the sample image to the trained font.

2.2 THRESHOLD



Thresholding is nothing more than separating pixels based upon their intensity value. For gray scale images, the range is from 0 to 255. Three automatic ROI methods of thresholding are included with the DLReader: computed, linear, and nonlinear.

Type Computed
Computed thresholding simply calculates a single value that is used across the entire ROI to extract pixels. This type of thresholding is relatively fast and works well when the lighting is uniform across the region of interest.

Type Linear & Block Size
Linear thresholding is used when one side is brighter than the other side and the light intensity changes uniformly from one side to the other. Used in conjunction with the value specified in the edit box labelled Block Size, linear thresholding calculates a value on the left side of the ROI, a value on the right side of the ROI, and then fills in the middle values in a linear fashion from left to right. The size of the steps from which to calculate the right and left side values is based on the value in the Block Size edit box.

Type Non Linear & Block Size
Non-linear thresholding divides the ROI into the number of blocks entered in the Block Size edit box, calculates a threshold value for each block, and then uses that value to extract pixel data. The minimum number of blocks is four (4) while the maximum number is fifty (50). As a rule, this value should be set to the same number of characters or objects to read. If there are spaces the width of a character or other kind of objects are present, increase the block count by one for each space. Non-linear thresholding is designed for applications where the lighting is uneven across the region of interest.

Threshold Values Low & High
These values are the output of the chosen thresholding method.

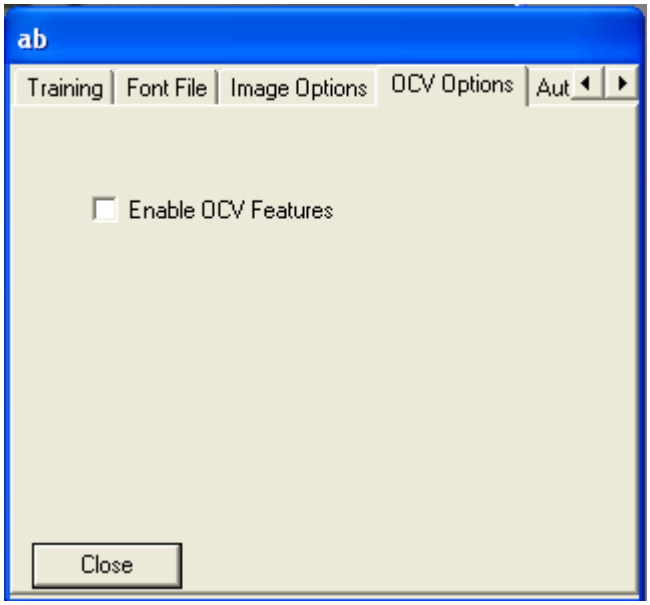
Integer Range
The *Integer Range* calculation changes the algorithm used to determine the threshold and to read the character. This will increase the speed of the operation so that reading with a calculated threshold is almost as fast as reading with a fixed threshold. Using the *Integer*

Range calculation sacrifices the accuracy when processing hard to read images. The accuracy on simpler images is not affected.
If Integer range is not checked, the *Floating Point* calculation is enabled.

Outline Char NOT USED

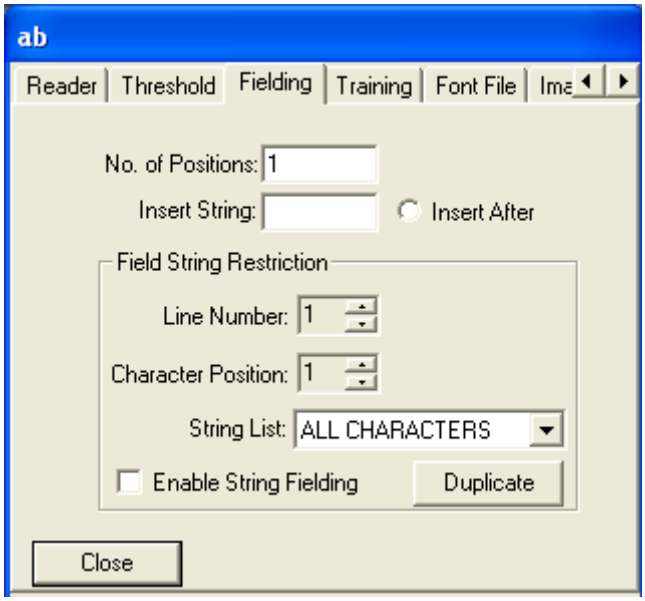
Gain& Offset
As a general rule, it is better to try each of the automatic threshold methods before adjusting gain and offset.
There is no time penalty for adjusting gain or offset. The calculation is made every time the threshold is computed.
Gain and offset is used is those special situations where the automatic thresholding algorithm does not correctly segment characters. The gain and offset are applied according to the following formula:
New threshold = (Calculated threshold * gain) + offset
In most cases, the offset is left at zero

2.3 OCV OPTION —NOT USED



This option is ignored.

2.4 FIELDING —NOT USED



Fielding is a means to restrict the trained font list by object position. Fielding can be used to improve accuracy and speed.

For example, if it is known that object position one can only contain a number, fielding can be used to tell the reader to use only numbers when attempting to match an object. This is especially important when using non-OCR font styles or reading degraded characters. Fielding can also serve to stop the reader when the Maximum Number of Characters is reached. In situations where the region of interest may include “noise” or artifacts to the right of the objects, this will reduce or eliminate the number of failed reads due to the inclusion of that artifact.

The **Line Number** fielding parameter restricts the search to a particular line number in a multiple read situation.

As the above figure shows, any position can be restricted to any one of the options of the String List: ALL CHARACTERS, ALPHANUMERIC, ALPHA ONLY, NUMERIC ONLY, HEXIDECIMAL, SPACE, FORCED CHAR, NON-ALPHANUMERIC, USER, PATTERN, 1st Letter of Month, 2nd Letter of Month, 3rd Letter of Month. Most are self-explanatory.
The **Space option** forces a space in the specified character position. This would be used to ignore a character.

The **Force Character option** works similarly to the Space option in that it ignores the character in a specified position, but instead of a space, it puts a user specified character in that position.

The option entitled **User** can be very useful in some situations. For example, if the ROI contains a numeric day of the month, the first position can be restricted to just 0, 1, 2, or 3. Position two will be restricted to the Numeric setting. By setting the field restrictions in this fashion, position one objects will only be compared to trained characters with the identifiers of 0, 1, 2, or 3, while skipping any others.

2.5 TRAINING

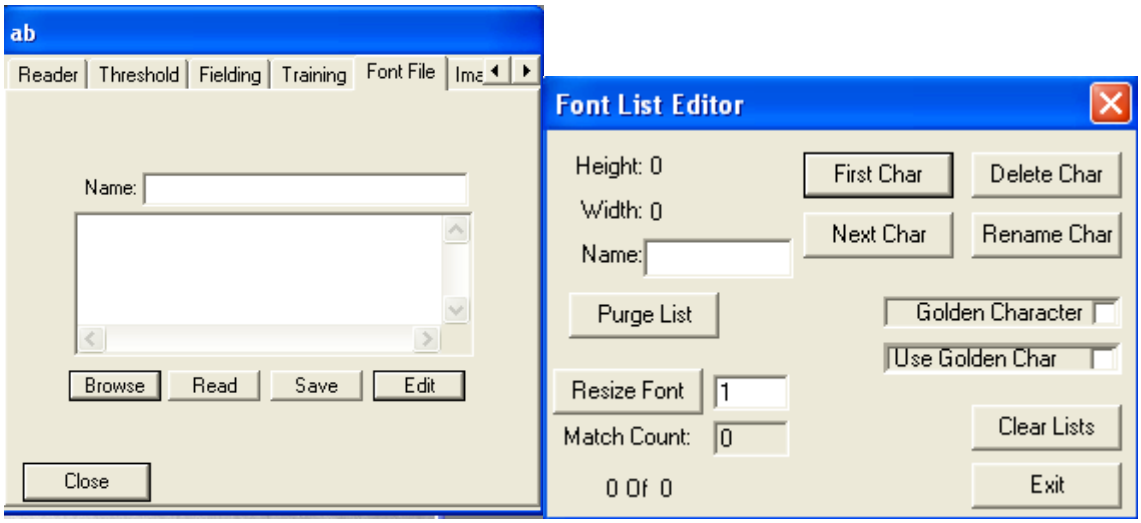
Once the characters or objects have been extracted with one bounding box for each character, the next logical step is training. Training the DLReader is very straightforward although enabling the training feature is a little more complex.

Enable Training
Click on the Enable Training check box on the Training tab page. Click on the Read Image button on the toolbar (or select OCR Read in the OCR menu) to bring up the training dialog box and type in one character for each object.

2.6 FONT FILE AND FONT LIST EDITOR

It is important to understand how the DLReader stores and uses trained characters. Unlike most other OCR readers, the DLReader stores data for **every** character trained! It is therefore unnecessary and **counterproductive** to train the same character from the same image more than once. When done, the identical character data is saved adding to storage requirements while slowing read performance.

In memory, a list of all trained character data is maintained in the order in which it was trained. When the DLReader attempts to read an object, **every** trained character is compared against the unknown object. Therefore, the longer the list of trained characters, the more the time needed to return the result. While there is a small penalty in time for this approach, a wide variety of the same character can be taught yielding very high read accuracies. If an object is not to be trained, type in the substitution character to skip that position. (The question mark – '?' – is the default substitution character.) Also, if only positions two and three in a triplet, for example, need training, enter the substitution character for the first position, the characters in the second and third position, and then click on the OK button. If fewer characters than the whole string need training, then fill in the positions up to and including just those objects and no more.



Name
Use the dialog box to name the font file.

Description
Optionally, a description in the Font Description box may be entered and saved with the font list. The description may be up to 255 characters in length.

Browse
To save a font, click on the Browse button to find a location to save the font file.

Read
Loading a saved font list is similar to saving it. Click on the Browse button. A file browse dialog box will appear to aid in locating the correct font file. Once located, click on the Read button to load the font file. If another font was being used with the presently open Reader, it may be desirable to delete the font file in memory so that it has no effect on the accuracy or speed. Clicking the Yes button in the inquiry box when it appears does this.

Save
After clicking on the Save button, an inquiry box will appear asking whether or not to delete the font file in memory. Clicking on *Yes* will erase the trained font from the volatile memory of the computer allowing the user to train a different font file without changing the settings on the Reader. This could be used if the image contained the entire list of characters for a particular font. The entire list could be trained and saved, then perhaps just the numbers could be trained for applications where only numbers are to be read and speed is essential.

Edit
If a mistake is made during the training process, the font list can be edited to either delete or rename any stored object. Open the font editor by clicking on the Edit button on the Font File tab page.

Font List Editor Parameters

This window can also be used to obtain the height and width of each object in the font list.

Name
Label associated with the trained object.

First Char
To rename or delete an object, click on the **First Char** button to start the font list, the image of the trained object will appear.

Delete Char
Click the **Next Char** button to scroll through the list to find the object.

Next Char
Then click the **Delete** button to remove the object from the list.

Rename Char
To rename the object, change the name in the **Name** edit box and click the **Rename** button.

Purge List
Erase all the trained Font.

Resize Font
Changes the trained Font dimension in order to fit a new Application constrain without re-train the Font.

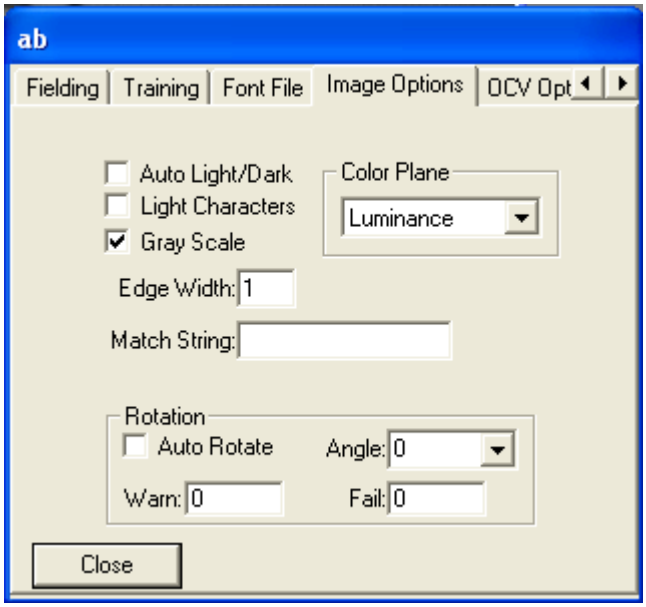
Clear Lists
Reset the counter that scroll through the list to find the object.

Golden Character
If more than one instance of the same object with the same Name is trained, one of them can be chosen as “Golden”: in this case the score of the recognized Label is calculated against the golden char even if it has matched best another instance.

Use Golden Char
Enables the use of the configured Golden Chars

2.7 IMAGE OPTION

The Image Option tab page allows the operator to set a number of parameters with respect to the image involved in the search.



Auto Light/Dark
This option will automatically determine if the objects are light on dark background or dark on light background. If checked, this option will be invoked on every read.

Light Characters
In the manual mode the option will set light characters.
In the automatic mode it will be used as feedback.

Gray Scale Image
Selects what kind of image data, gray scale or color, is to be used.
ONLY GRAY SCALE Enabled IS ALLOWED.

Color Plane
Used to select appropriate color for reading. NOT USED.

Edge Width
Number of consecutive pixels within the threshold range that define a valid edge. NOT USED.

Match String
String that the OCR check after reading.

Auto Rotate
When checked the SW tests both the position Angle and Angle+180.

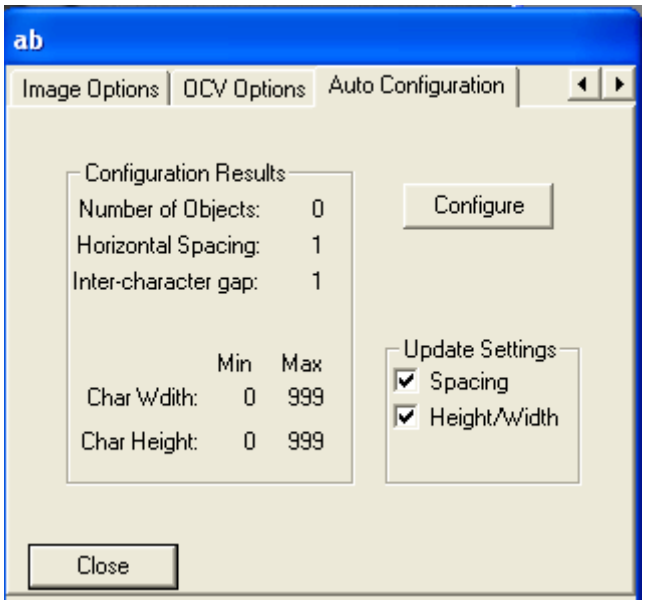
Angle
Ordinal position of the object, 0, 90, 180 or 270 degrees.

Warn
Warning Quality Level 1-1000. If it's value is meaningful (not 0), the OCR library raises a WARNING error that can be used inside the MATRIX application to warn the user about the decreasing quality.

Fail
Failure Quality Level 1-1000. If it's value is 0 (default), the char's scoring is controlled only by the acceptance value.

2.8 AUTO CONFIGURATION

In most cases Auto Configuration works well for setting object parameters, however in some cases it may return erroneous data. Auto Configuration will affect Reader tab page options.



Configure
Auto Configuration is a method of obtaining a starting point for setting the parameters for the size and spacing of objects that will be searched for in the character recognition process. For any given ROI, Auto Configuration will determine the number of characters in the string, the horizontal spacing and the inter-character gap. It will also compute the minimum and maximum character widths and heights. These values are all reported in the Auto Configuration tab page when an ROI box has been placed tightly around an object string and the Configure button has been pushed.

Number of Objects
Auto Configuration will return the number of objects found in the string if the read takes place successfully.

Horizontal Spacing
Horizontal spacing is a pixel value and defines how far apart dots or pieces of a character can be located. The Configure function returns this value.

Inter-Character Gap
This value is the minimum horizontal spacing between adjacent characters. The default value is one (1), and is retuned by the Configure function.

Char Width
Minimum and maximum values for character width are returned by the Configure function.

Char Height
Minimum and maximum values for character height are returned by the Configure function.

2.9 SAVING AND RESTORING READER SETTINGS

The entire setup, including the position, shape and size of the ROI can be saved for later use. This is useful for sharing the information or simply performing a read operation on an image at a later time.

To save an image, click on the Save Readers button on the toolbar, or select Save Reader Data from the OCR menu. Enter a file name in the dialog box and click Save. The parameters that are saved are identified by the checkboxes on the Save Options tab page.

Saved settings can easily be restored. Once the DLReader is started and an image is opened, it is not necessary to create a reader if saved settings are to be restored. Click on the Restore Readers button on the toolbar or select Restore Reader Data from the OCR menu to restore the saved reader(s). It must be remembered that the font file will be saved in the location indicated on the Font File tab page. If the information is shared with another computer, the file must be in that location for the DLReader to find the file.

Saved readers can be restored at any time. They will replace any reader or readers that may be in use at the time of restoration. If a reader is restored and the settings are changed, it is easy to return to the original saved settings. Simply restore the saved reader.

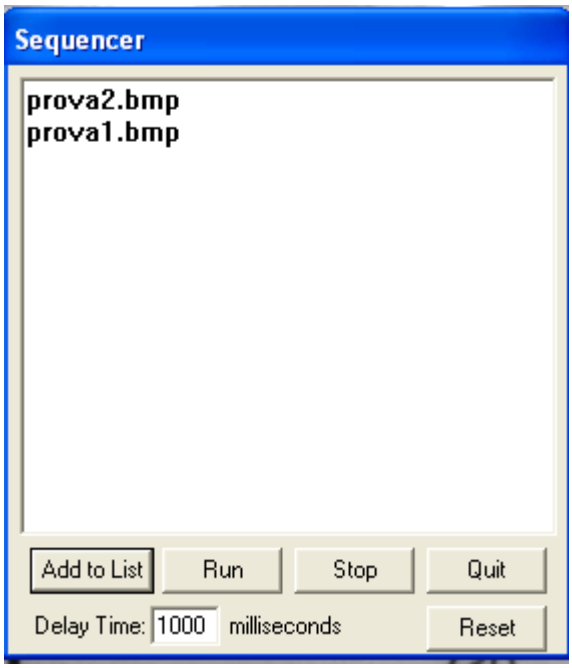
2.10 CHANGING IMAGES

When using the DLReader, there are three (3) ways to change the images: close the image, replace the image or run a sequence of images.

In order for the DLReader to operate, it is necessary to have an open image.

Closing the active image closes all readers and all the unsaved setting information is lost. If the active image is the only open image, the DLReader is returned to its original state.

To use the reader or readers on a new image, the original image must be replaced. Select Replace Image from the File menu. This will allow the user to import a new image to be used with the same reader. All aspects of the reader, including the ROI, will remain unchanged. If a series of images are to be used, replacing the image will set the new image in place of the previous image. If the characters in the new image appear within the ROI, the user needs only click the Read Image button to read the characters.



To read a series of images, it is possible to use the Sequence feature. Select Sequence from the OCR menu. Click the Add to List button on the Sequencer box to add a string of images. In the Open dialog box, the images can be selected. All the images must be located in the same file folder. By holding the CTRL button down, all the images may be selected. Clicking on the Run button will start the sequencer. The delay between images can be set with the Delay Time edit box.

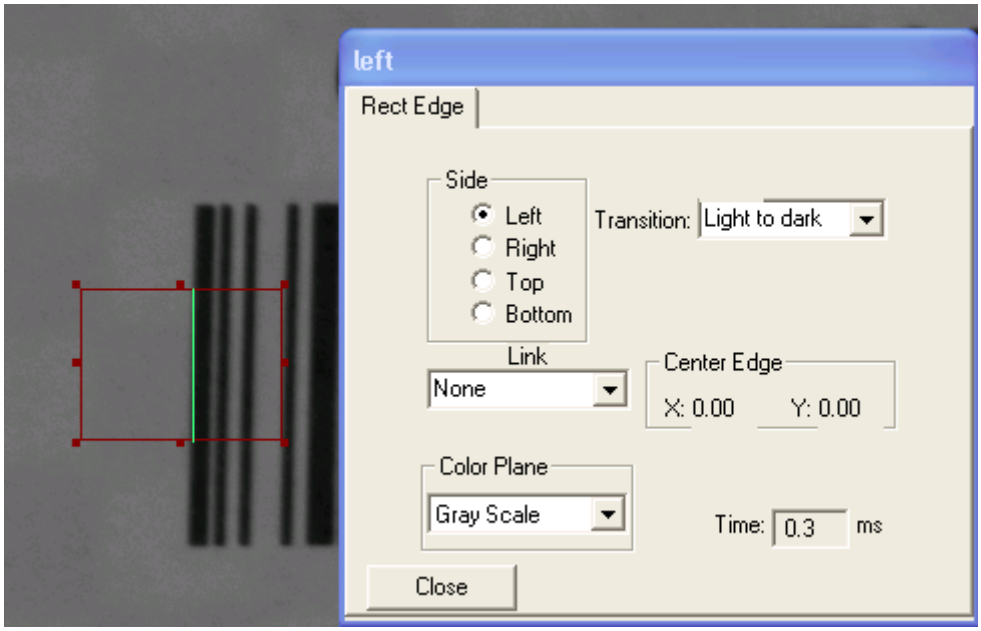
The sequencer is a good feature for observing the DLReader in full operation. Used with the search feature, each character string can be located and read demonstrating all the features of the DLReader.

3 EDGE FINDERS

The first, and most important part of any OCR effort is finding the characters to be read.

In industrial applications, it is normal that the marking will be done in a specific location on a surface. There will no doubt be variation in the location of the marking, but the marking will not be in a random location. This means other things, such as marking or label edges, can be used to quickly find the characters of interest. It is not necessary that the edge be continuous or in any specific area with respect to the characters of interest. It is also possible for the characters themselves to be used by the Edge Finder to find the marking to be read.

A linking feature has been included so that the relative position of the found edge and the ROIs of additional readers can be maintained. In some cases, multiple Edge Finders are set up and linked so that the first will find an edge and reposition the next to find another edge. The Edge Finders can be linked in series or parallel or a combination of both to get the best positioning of the character reading ROI.



3.1 CREATING AN EDGE ROI

With the last button of the “OCR” menu bar an Edge ROI is created.

⇒ The user is prompted to choose a name (unique) to give to the ROI.

⇒ The user’s needs to correctly position it on the edge to be found.

NOTE:
The only supported Edge ROIs are rectangular and can be configured to find only horizontal/ vertical edges.

3.2 FINDING THE EDGE

The first step in finding the edge is determining which edge is to be used.
Selecting the **Side** radio button will determine how the search for the edge will be conducted.

For example, choosing the Right side will cause the program to start from the right hand side of the ROI and search to the left for an edge.

The **Transition** control allows the choice of a transition from light to dark, dark to light or any edge transition.

3.3 OUTPUT INFORMATION (SHOWN ONLY GRAPHICALLY)

The Edge Finder gives information to the user that can help with setting up an OCR application that might use this tool.

The **Center Edge** gives the center location of the found edge, as the name would indicate. This value is not updated in the DLReader Edge window but it is only shown in the Edge ROI.

The **Time** is the reading time of the Edge Finder.